

Examen Parcial de Programación II – Ejercicio Práctico

17 de Mayo de 2015 – HOJA DE RESPUESTA

Apellidos _____

Nombre _____

DNI/NIE/Pasaporte _____ N° Matrícula _____

Clase CuentaDeposito: Cabecera de la clase, atributo adicional, constructor, métodos ingresar(), retirar() y marcarVencido().

```
// Cuenta bancaria de depósito (solo permite movimientos tras finalizar
// el plazo, y no permite descubiertos)
public class CuentaDeposito extends Cuenta {
    private boolean vencido;

    // Constructor
    public CuentaDeposito(String titular, int saldoInicial)
        throws ExcepcionOperacionNoValida {
        super(titular, saldoInicial);
        vencido = false;
    }

    // No se permiten ingresos
    @Override
    public void ingresar(int importe) throws ExcepcionOperacionNoValida {
        throw new ExcepcionOperacionNoValida();
    }

    // No se permiten descubiertos ni retiradas antes del plazo
    @Override
    public void retirar(int importe) throws ExcepcionOperacionNoValida {
        if (!vencido || importe > getSaldo())
            throw new ExcepcionOperacionNoValida();
        super.retirar(importe);
    }

    // Marcar que el depósito ha vencido (se ha cumplido el plazo)
    public void marcarVencido() {
        vencido = true;
    }
}
```

Método saldoTotal(titular) de la clase Banco.

```
// Consultar saldo total
public int saldoTotal(String titular) {
    int total = 0;
    for (int k = 0; k < cuentas.length; k++) {
        Cuenta c = cuentas[k];
        if (c != null && c.getTitular().equals(titular)) {
            total = total + c.getSaldo();
        }
    }
    return total;
}
```

Método ingresar(importe, titular) de la clase Banco.

```
// Realizar ingreso
public void ingresar(int importe, String titular)
    throws ExcepcionOperacionNoValida {
    boolean hecho = false;
    for (int k = 0; k < cuentas.length && !hecho; k++) {
        Cuenta c = cuentas[k];
        if (c != null && c.getTitular().equals(titular) &&
            !(c instanceof CuentaDeposito)) {
            c.ingresar(importe);
            hecho = true;
        }
    }
    if (!hecho) {
        throw new ExcepcionOperacionNoValida();
    }
}
```

```
// Realizar ingreso - SOLUCION ALTERNATIVA
public void ingresar(int importe, String titular)
    throws ExcepcionOperacionNoValida {
    boolean hecho = false;
    for (int k = 0; k < cuentas.length && !hecho; k++) {
        Cuenta c = cuentas[k];
        if (c != null && c.getTitular().equals(titular)) {
            try {
                c.ingresar(importe);
                hecho = true;
            } catch (ExcepcionOperacionNoValida e) {
                // ignorarlo, seguir intentando
            }
        }
    }
    if (!hecho) {
        throw new ExcepcionOperacionNoValida();
    }
}
```